

Juno v.0.4.0: Production-ready

Nethermind's Starknet client implementation is ready for production use.

[Rian Hughes](#)

[Follow](#)

Nethermind.eth

--

Listen

Share

By

[Rian Hughes

](<https://www.linkedin.com/in/rianhughes/>)and

[Jorik Schellekens

](<https://twitter.com/mempoolsurfer>).

We're excited to introduce the latest version of Juno — Juno 0.4!

This new release marks a significant milestone for Juno, as it has now reached a production-ready status and offers complete support for the RPC specification.

Engineered with a focus on reliability and performance, Juno 0.4 streamlines your interaction with the Starknet network, making it an ideal choice for developers and businesses alike. Join us in exploring this update and its potential to elevate your Starknet operations.

With the completion of the final RPC methods, Juno can serve as the RPC for any wallet, explorer, indexer, or developer.

What is included in this release?

- The final outstanding Starknet RPC Methods

:

- starknet_call
- starknet_estimateFee
- starknet_addDeclareTransaction
- starknet_addDeployAccountTransaction
- starknet_addInvokeTransaction
- Juno-specific RPC methods:
- juno_getTransactionStatus
- juno_version
- L1 Verifier

: Verification of state from Layer 1 has been implemented.

- Block Reorg Detection and Handling

: A feature to detect and handle block reorganizations has been implemented.

- gRPC Service

: To accommodate users requiring direct access to the database, a gRPC service has been exposed.

- Database Migration

: The system has been improved to handle database changes more gracefully. It's no longer necessary to sync from the start when some database changes occur.

- Starknet v0.12.0 support:

The Rust VM has been integrated, meaning Juno can now execute transactions.

In addition to these features, we also provide Snapshots of synced Juno DBs to accelerate syncing. Note that this state can be trusted because the state root is verified against the L1 commitment.

RPC support

Juno now supports RPC methods to add "Invoke", "Declare", and "DeployAccount" transactions to the network. Currently, these methods relay the transaction to the Sequencer, although in the future, they will also add them into a local mempool. Supporting these transactions now is an important step in helping to replace the centralized Sequencer with a distributed network of nodes, a crucial milestone in the decentralization of Starknet.

In addition, we've introduced support for two other RPC methods: `starknet_call`

, and `starknet_estimateFee`

, which allow users to simulate and estimate the fee of their transactions. Importantly, this can be done without submitting a state-changing transaction to the network; it's all done locally within the Juno client thanks to the new integration of a Rust VM.

The remaining Juno-specific RPC methods will allow users to track their transactions along their lifecycle and know which version of Juno they are interacting with.

Performance

The Juno team is creating an efficient full-node implementation. For detailed information on the current statistics of Juno, explore the [Juno Full Node Implementation Performance

](<https://nethermind.notion.site/Juno-Full-Node-Implementation-Performance-918a7828fd254be29614ebe725613c6e>) page. Additionally, we are working on a dashboard that automates the tests and benchmarks of the full-node releases.

The following metrics provide insights into the performance of Juno during a full sync of the network from genesis.

The Juno team is working on reducing the sync time as much as possible. With the latest [v0.4.1 release](#), significant improvements have been made to reduce sync times, resulting in a 1.7x reduction. Specifically, the time required to sync 90,000 blocks has been reduced from 41 hours to 24 hours. Juno plans to further improve these sync times with the next two major roadmap tickets.

The road ahead

With the 0.12 release of Starknet, the network performance improved tremendously. The future releases of Starknet, 0.14 and 0.15, will focus on usability improvements for the network.

Meanwhile, Juno now has two missions:

1. Sync much faster than the impressive new speed of the network
2. Make practical steps toward decentralization.

To further improve sync times and reduce the computational resources required to participate in the network, Juno plans to implement "snap-sync". This will allow nodes to request the full state according to the root at a recent block, verify this state's correctness, and rapidly sync the remaining blocks to the head of the chain. To do this, a full node must serve the state-Trie to peers, which requires iterating over the leaves of the entire state-Trie. Juno's path-based state-Trie makes iterating over, and hence serving, the leaves of the state-Trie very efficient!

The Juno team is currently testing how performant the process of building the state-Trie is from just the leaves. This knowledge will allow us to make predictions about the speed of the sync.

In collaboration with Pathfinder, we're implementing a peer-to-peer (p2p) network for full nodes to sync data from the Starknet network. Establishing a distributed network of full nodes is a crucial step in Starknet's decentralization journey,

allowing the centralized Sequencer to go offline in the future. A p2p network will also improve sync times as nodes can sync from multiple peers simultaneously. The p2p protocol is currently being tested in a small testnet environment.

Get involved!

Having completed the implementation of the final RPC methods, Juno now serves as a reliable RPC option for wallets, explorers, indexers, and developers. Moreover, Juno's low memory requirements make it an ideal client if you're looking to build data solutions for Starknet. Please reach out to our engineering team to help you get started with infra and configurations.

Finally, we encourage everyone in the community, whether you're a seasoned developer, blockchain enthusiast, or simply curious about the project, to engage with Juno. Our door is always open for questions, discussions, and feedback.

Get involved with Juno to help shape its evolution and make a direct impact on the future of Starknet. Remember, Juno isn't just our project; it belongs to the entire community.

About Us

Nethermind is a team of world-class builders and researchers. We empower enterprises and developers worldwide to access and build upon the decentralized web. Our work touches every part of the web3 ecosystem, from our Nethermind node to fundamental cryptography research and infrastructure for the Starknet ecosystem.

If you're interested in solving some of blockchain's most difficult problems, visit our

[\[job board](#)

](<https://nethermind.page.link/careers>)!

Disclaimer

This post has been prepared for the general information and understanding of the readers. No representation or warranty, express or implied, is given by Nethermind as to the accuracy or completeness of the information or opinions contained in the above post. In no event will Nethermind be liable for any loss or damage, including without limitation, indirect or consequential loss or damage, arising out of or in connection with the use of any information provided in this blog post.